

On the Complexity of Disjunction and Explicit Definability Properties in Some Intermediate Logics

Mauro Ferrari¹, Camillo Fiorentini¹, and Guido Fiorino²

¹ Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Via Comelico 39, 20135 Milano, Italy

² CRII, Università dell'Insubria, Via Ravasi 2, 21100 Varese, Italy

Abstract. In this paper we provide a uniform framework, based on extraction calculi, where to study the complexity of the problem to decide the disjunction and the explicit definability properties for Intuitionistic Logic and some Superintuitionistic Logics. Unlike the previous approaches, our framework is independent of structural properties of the proof systems and it can be applied to Natural Deduction systems, Hilbert style systems and Gentzen sequent systems.

1 Introduction

In recent years there has been a growing interest in studying the complexity of intuitionistic proofs, in particular in connection with the decision of the disjunction property (DP) and the explicit definability property (ED) [2,3]. Formally, given a logic \mathbf{L} satisfying (DP) and (ED), deciding (DP) means to find out which between A and B is provable in \mathbf{L} given a proof of $A \vee B$. Analogously, deciding (ED) means to find out a term t such that $A(t)$ is provable in \mathbf{L} given a proof of $\exists x A(x)$. On the propositional side, in [2,3] it is shown that the disjunction property for Intuitionistic Logic can be decided in polynomial time in the size of the proof of $A \vee B$. In [2] the result is based on proofs of a Natural Deduction Calculus, while the result of [3] is based on proofs of a Sequent Calculus and is extended to the case of Harrop assumptions. We remark that in both cases the result essentially depends on structural properties of the proofs of the calculus in hand. As for the case of Intuitionistic first-order Logic, in [2] it is shown that (ED) can be decided in exponential time in the size of the proof of $\exists x A(x)$ for languages without function symbols, and in superexponential time for the full language; moreover, for the latter case a superexponential lower-bound is provided.

In this paper we introduce a uniform framework for studying the complexity of (DP) and (ED) in Intuitionistic and Superintuitionistic Logics. The main difference between our approach and those of [2,3] is that we define an explicit calculus, we call *extraction calculus*, to analyze the information contained into a proof, and we exploit it to decide (DP) and (ED). The extraction calculus uses as axioms all the sequents that can be extracted from a proof and some “simple”

inference rules. Here the term “simple” mainly refers to the intuitionistic nature of these rules. As a matter of fact, the main rule of our extraction calculi is an inference rule formalizing SLD-resolution. To treat the case of Harrop assumptions restricted versions of \wedge -elimination, \rightarrow -elimination and \forall -elimination are required. Finally, we show that also to treat two well-known Superintuitionistic Logics, namely Kuroda Logic and Grzegorczyk Logic, intuitionistic rules are sufficient. More than this, also the complexity of (DP) and (ED) in these systems is the same as Intuitionistic Logic. We point out that, although in this paper we apply the extraction calculus to Natural Deduction proofs, our framework is independent of the nature of the proof systems, indeed it can be applied to Natural Deduction systems, Hilbert style systems and Gentzen sequent systems, while the techniques of [2,3] depend on structural properties of the calculi in hand.

2 Preliminaries and Extraction Calculi

The set of *terms* and the set of (first-order) *formulas* of the language \mathcal{L} are built up in the usual way starting from a denumerable set of individual variables, an extra-logical alphabet \mathcal{A} , and the logical constants $\perp, \wedge, \vee, \rightarrow, \forall, \exists$; moreover, we consider $\neg A$ as an abbreviation for $A \rightarrow \perp$. The notion of first-order *substitution* is the usual one. A *sequent* is any expression of the form $\Gamma \vdash A$, where A is a formula and Γ is a finite set of formulas; when Γ is empty we simply write $\vdash A$.

In the sequel, we introduce the extraction calculus we use to decide disjunction property (DP) and explicit definability property (ED) for some logics. We remark that, although in this paper we apply the extraction calculus to Natural Deduction proofs, in its general formulation it can be applied to a great variety of calculi. For this reason the formulation of the extraction calculus is based on an abstract notion of proof and calculus (for a complete discussion we refer the reader to [4,5,7]).

A *proof* over \mathcal{L} is any finite object π such that:

1. The (finite) set of formulas of \mathcal{L} occurring in π is uniquely determined and nonempty;
2. π proves a sequent $\Gamma \vdash A$, where Γ (possibly empty) is the set of *assumptions* of π , while A is the *consequence* of π .

The notation $\pi : \Gamma \vdash A$ means that $\Gamma \vdash A$ is the sequent proved by π . The *size* of a *proof* is the number of symbols occurring in the proof, where a symbol is an occurrence of a constant, an individual variable, a predicate symbol, a logical constant.

A *calculus* over \mathcal{L} is a pair $(\mathbf{C}, [\cdot])$, where \mathbf{C} is a recursive set of proofs over \mathcal{L} and $[\cdot]$ is a recursive map associating with every proof of the calculus the set of its subproofs. We require $[\cdot]$ to satisfy the following natural conditions: $\pi \in [\pi]$ and, for every $\pi' \in [\pi]$, $[\pi'] \subseteq [\pi]$. We remark that any usual single conclusion inference system is a calculus according to our definition. With an abuse of notation we often identify a calculus $(\mathbf{C}, [\cdot])$ with the set \mathbf{C} of its proofs.

Given $\Pi \subseteq \mathbf{C}$, $\text{Seq}(\Pi) = \{\Gamma \vdash A \mid \pi : \Gamma \vdash A \in \Pi\}$ is the set of the *sequents proved in Π* and $[\Pi] = \{\pi' \mid \text{there exists } \pi \in \Pi \text{ such that } \pi' \in [\pi]\}$ is the *closure under subproofs* of Π in the calculus \mathbf{C} .

Let R be an inference rule of the kind

$$\frac{\Gamma_1 \vdash A_1 \quad \dots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}_R$$

R is an *extraction rule (e-rule)* for \mathbf{C} if:

- R is an admissible rule in \mathbf{C} , that is $\{\Gamma_1 \vdash A_1, \dots, \Gamma_n \vdash A_n\} \subseteq \text{Seq}(\mathbf{C})$ implies $\Delta \vdash B \in \text{Seq}(\mathbf{C})$;
- R can be polynomially simulated in \mathbf{C} . That is, there exists a polynomial time algorithm in the size of the input proofs that, given $\pi_1 : \Gamma_1 \vdash A_1, \dots, \pi_n : \Gamma_n \vdash A_n$ in \mathbf{C} , builds a proof $\pi : \Delta \vdash B$ in \mathbf{C} .

We point out that the above definition of e-rule is different from the one given in [4,5,7] where the authors consider the logical complexity of extraction calculi instead of their computational complexity.

Definition 1 (Extraction Calculus). *Given a set \mathcal{R} of e-rules for \mathbf{C} and a recursive set $\Pi \subseteq \mathbf{C}$, the extraction calculus for Π , denoted by $\text{ID}(\mathcal{R}, [\Pi])$, is defined as follows:*

1. *If $\Gamma \vdash A \in \text{Seq}([\Pi])$, then*

$$\tau \equiv \frac{}{\Gamma \vdash A}$$

is a proof-tree of $\text{ID}(\mathcal{R}, [\Pi])$ and τ proves $\Gamma \vdash A$.

2. *If $\tau_1 : \Gamma_1 \vdash A_1, \dots, \tau_n : \Gamma_n \vdash A_n$ are proof-trees of $\text{ID}(\mathcal{R}, [\Pi])$ and*

$$\frac{\Gamma_1 \vdash A_1 \quad \dots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}_R$$

is a rule of \mathcal{R} , then the proof-tree

$$\tau \equiv \frac{\tau_1 : \Gamma_1 \vdash A_1 \quad \dots \quad \tau_n : \Gamma_n \vdash A_n}{\Delta \vdash B}_R$$

belongs to $\text{ID}(\mathcal{R}, [\Pi])$ and τ proves $\Delta \vdash B$.

When Π consists of a single proof π , we simply denote the extraction calculus with $\text{ID}(\mathcal{R}, [\pi])$.

In the sequel we consider Intuitionistic and Superintuitionistic logics. We denote with **Int** the set of intuitionistically valid formulas of the pure first-order language \mathcal{L} . In Table 1 we give the rules of the Natural Deduction calculus $\mathcal{ND}_{\text{Int}}$ for first-order Intuitionistic Logic of [17]. A proof π of $\mathcal{ND}_{\text{Int}}$ is a tree of sequents built using the rules of Table 1. The sequent proved by π is the lowest sequent of π and the notions of subproof of π and $\text{depth}(\pi)$ are defined in the obvious way. Hereafter we assume the usual conventions on *proper parameters* and *free variables* of the natural deduction rules stated in [17] in such a way to guarantee

Table 1. The Natural Deduction calculus $\mathcal{ND}_{\mathbf{Int}}$ for Intuitionistic Logic

$\frac{}{\Gamma \vdash A} \text{Id}$	$\frac{\Gamma \vdash \perp}{\Gamma \vdash A} \perp_{\text{Int}} \text{ where } A \text{ is an atomic formula.}$
$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \wedge_{\text{I}}$	$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge_{\text{E}}$ $\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge_{\text{E}}$
$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee_{\text{I}}$	$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vee_{\text{I}}$ $\frac{\Gamma \vdash A \vee B \quad \Delta, A \vdash C \quad \Theta, B \vdash C}{\Gamma, \Delta, \Theta \vdash C} \vee_{\text{E}}$
$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_{\text{I}}$	$\frac{\Gamma \vdash A \quad \Delta \vdash A \rightarrow B}{\Gamma, \Delta \vdash B} \rightarrow_{\text{E}}$
$\frac{\Gamma \vdash A(y/x)}{\Gamma \vdash \forall x A(x)} \forall_{\text{I}} \text{ where } y \text{ does not occur free in } \Gamma \text{ or } \forall x A(x).$	$\frac{\Gamma \vdash \forall x A(x)}{\Gamma \vdash A(t/x)} \forall_{\text{E}}$
$\frac{\Gamma \vdash A(t/x)}{\Gamma \vdash \exists x A(x)} \exists_{\text{I}}$	$\frac{\Gamma \vdash \exists x A(x) \quad \Delta, A(y/x) \vdash C}{\Gamma, \Delta \vdash C} \exists_{\text{E}} \text{ where } y \text{ does not occur free in } \Delta, \exists x A(x) \text{ or } C.$

that the tree-structure $\theta\pi$, obtained by replacing every free variable x occurring in π with the term $\theta(x)$, is a well-defined proof.

To conclude this section we notice that extraction calculi have been introduced in [4,5,7] to define a class of systems for which (DP) can be decided using only information contained in a proof of $A \vee B$ (the same holds for (ED)). Such a class contains formal systems that cannot be treated with Normalization, Cut-elimination or Realizability. Extraction calculi have also been applied in the framework of program synthesis from formal proofs, see [1,6].

3 Propositional Intuitionistic Logic

In this section we consider the case of propositional Intuitionistic Logic. We denote with \mathcal{L}_p the propositional fragment of \mathcal{L} , with \mathbf{Int}_p the propositional Intuitionistic Logic and with $\mathcal{ND}_{\mathbf{Int}_p}$ the calculus consisting of the propositional rules of Table 1. It is well-known that \mathbf{Int}_p meets (DP) and in [2,3] it is proved that (DP) can be decided in polynomial time in the size of a proof of $A \vee B$. Here we show an analogous result obtained with a different technique. In particular, given a proof $\pi : \vdash A \vee B$ of $\mathcal{ND}_{\mathbf{Int}_p}$ we exhibit an algorithm to construct a proof of $\vdash A$ or a proof of $\vdash B$ of $\mathcal{ND}_{\mathbf{Int}_p}$ in polynomial time in the size of π , using an

extraction calculus. Let us consider the following inference rule formalizing SLD resolution in the propositional setting:

$$\frac{\vdash A_1 \quad \dots \quad \vdash A_n \quad \overline{A_1, \dots, A_n \vdash B}}{\vdash B} \text{SLD}_p$$

where A_1, \dots, A_n, B are arbitrary formulas. It is easy to check that SLD_p is an e-rule for $\mathcal{ND}_{\text{Int}_p}$. Now, we show that such a rule is enough to decide the disjunction property for intuitionistic formulas. To this aim we introduce the following notion of evaluation:

Definition 2 (Propositional Evaluation). *Given a set of proofs Π of a calculus \mathbf{C} and a formula A , A is evaluated in Π (in symbols $\Pi \triangleright A$) iff the following conditions hold:*

- (i). *There exists a proof $\pi : \vdash A \in \Pi$.*
- (ii). *One of the following inductive conditions holds:*
 - a) *A is an atomic or a negated formula;*
 - b) *$A \equiv B \wedge C$ and $\Pi \triangleright B$ and $\Pi \triangleright C$;*
 - c) *$A \equiv B \vee C$ and either $\Pi \triangleright B$ or $\Pi \triangleright C$;*
 - d) *$A \equiv B \rightarrow C$ and, if $\Pi \triangleright B$, then $\Pi \triangleright C$.*

A set Γ of formulas is evaluated in Π (and we write $\Pi \triangleright \Gamma$) if $\Pi \triangleright A$ holds for every $A \in \Gamma$.

Lemma 1. *Let Π be a recursive set of proofs of $\mathcal{ND}_{\text{Int}_p}$. For every $\pi : \Gamma \vdash A$ belonging to $[\Pi]$, if $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright \Gamma$ then $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright A$.*

Proof. Since $\Gamma = \{B_1, \dots, B_n\}$ is evaluated in $\text{ID}(\text{SLD}_p, [\Pi])$, there exist in $\text{ID}(\text{SLD}_p, [\Pi])$ the proofs $\tau_1 : \vdash B_1, \dots, \tau_n : \vdash B_n$. Moreover, since $\Gamma \vdash A \in \text{Seq}([\Pi])$, the proof

$$\frac{\tau_1 : \vdash B_1, \dots, \tau_n : \vdash B_n \quad \overline{\Gamma \vdash A}}{\vdash A} \text{SLD}_p$$

belongs to $\text{ID}(\text{SLD}_p, [\Pi])$. This proves Point (i) of Definition 2; to prove Point (ii) we proceed by induction on $\text{depth}(\pi)$.

Basis: If $\text{depth}(\pi) = 0$, the only rule applied in π is an assumption introduction Id, hence $\Gamma = \{A\}$ and the assertion trivially holds.

Step: Let us suppose that $\text{depth}(\pi) = h + 1$. The proof goes on by cases according to the last rule applied in π ; here we only discuss some representative cases.

Disjunction Elimination.

$$\pi : \Gamma \vdash A \equiv \frac{\pi_0 : \Gamma_0 \vdash B_1 \vee B_2 \quad \pi_1 : \Gamma_1, B_1 \vdash A \quad \pi_2 : \Gamma_2, B_2 \vdash A}{\Gamma_0, \Gamma_1, \Gamma_2 \vdash A} \text{E}\vee$$

Since $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright \Gamma_0$, π_0 belongs to $[\Pi]$ and $\text{depth}(\pi_0) \leq h$, we get, by induction hypothesis, that $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright B_1 \vee B_2$. Thus, there exists $i \in \{1, 2\}$ such that $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright B_i$ and, since $\pi_i : \Gamma_i, B_i \vdash A$ belongs to $[\Pi]$, by induction hypothesis we have $\text{ID}(\text{SLD}_p, [\Pi]) \triangleright A$.

Implication Introduction.

$$\pi : \Gamma \vdash A \equiv \frac{\pi' : \Gamma, B \vdash C}{\Gamma \vdash B \rightarrow C} \mapsto$$

Let us assume that $\text{ID}(\text{SLD}_p, [II]) \triangleright B$; since $\text{ID}(\text{SLD}_p, [II]) \triangleright \Gamma$, π' belongs to $[II]$ and $\text{depth}(\pi') \leq h$, by induction hypothesis we get $\text{ID}(\text{SLD}_p, [II]) \triangleright C$. \square

Since the empty set of formulas is trivially evaluated in $\text{ID}(\text{SLD}_p, [\pi])$, by the above lemma we deduce that, if $\pi : \vdash A \vee B$ is a proof of $\mathcal{ND}_{\text{Int}_p}$, then $\text{ID}(\text{SLD}_p, [\pi]) \triangleright A \vee B$. Hence, $\text{ID}(\text{SLD}_p, [\pi]) \triangleright A$ or $\text{ID}(\text{SLD}_p, [\pi]) \triangleright B$ and, by Point (i) of Definition 2, one between the sequents $\vdash A$ and $\vdash B$ has a proof in $\text{ID}(\text{SLD}_p, [\pi])$. Hence:

Theorem 1. *If $\pi : \vdash A \vee B$ is a proof of $\mathcal{ND}_{\text{Int}_p}$, then either $\vdash A$ or $\vdash B$ is provable in $\text{ID}(\text{SLD}_p, [\pi])$.*

Now, to study the complexity of the disjunction property, let us introduce the following map. Let SEQ_p be the set of all the sequents over \mathcal{L}_p ; given a finite set of sequents Σ , the function $E_\Sigma : 2^{\text{SEQ}_p} \rightarrow 2^{\text{SEQ}_p}$ is defined as follows:

$$E_\Sigma(\Delta) = \{ \vdash A \mid B_1, \dots, B_n \vdash A \in \Sigma \text{ and } \{ \vdash B_1, \dots, \vdash B_n \} \subseteq \Delta \}$$

It is easy to check that E_Σ is a monotone and continuous operator on the complete partial order $\langle 2^{\text{SEQ}_p}, \subseteq \rangle$. Hence, by the Knaster-Tarsky Theorem, E_Σ has the least fixpoint E_Σ^∞ and, by Kleene Theorem, $E_\Sigma^\infty = \bigcup_{k \in \omega} E_\Sigma^k$, where:

$$\begin{aligned} E_\Sigma^0 &= \emptyset \\ E_\Sigma^{k+1} &= E_\Sigma(E_\Sigma^k) \end{aligned}$$

Given a finite set of proofs Π of $\mathcal{ND}_{\text{Int}_p}$, let $\Sigma = \text{Seq}([II])$; it is immediate to check that $\vdash A \in E_\Sigma^\infty$ iff $\vdash A$ is provable in $\text{ID}(\text{SLD}_p, [II])$.

Theorem 2. *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Int}_p}$, there exists a polynomial time algorithm that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\text{ID}(\text{SLD}_p, [\pi])$.*

Proof. By Theorem 1 we know that either a proof of $\vdash A$ or a proof of $\vdash B$ belongs to $\text{ID}(\text{SLD}_p, [\pi])$; this implies that, setting $\Sigma = \text{Seq}([\pi])$, either $\vdash A$ or $\vdash B$ belongs to E_Σ^∞ . Thus, we have to generate the sequence of the E_Σ^k ($k \geq 0$) until one between the above sequents will occur. We point out that $|E_\Sigma^\infty| \leq n$, where n is the number of sequents of Σ , hence each E_Σ^k contains at most n sequents. Since at any application of E_Σ we add at least a new sequent (otherwise E_Σ^k is already the fixpoint), we need at most n iterations to generate E_Σ^∞ . At iteration $i+1$ we have to consider any sequent $B_1, \dots, B_m \vdash A$ in Σ not already used and check whether $\vdash B_1, \dots, \vdash B_m$ belong to E_Σ^i , in this case we add $\vdash A$ to E_Σ^{i+1} and we mark the sequent $B_1, \dots, B_m \vdash A$ as used. Clearly, this can be done in polynomial time in the size of π . Since each iteration can be performed in polynomial time, the procedure has polynomial time complexity in the size of π . Finally, for any sequent in E_Σ^{i+1} we can retrieve the sequence of SLD_p rules applied to derive it, hence we can build the proof of $\vdash A$ or $\vdash B$ in polynomial time. \square

Since SLD_p is polynomially simulable in $\mathcal{ND}_{\text{Int}_p}$, any proof of $\text{ID}(\text{SLD}_p, [\pi])$ can be translated into a proof of $\mathcal{ND}_{\text{Int}_p}$ in polynomial time in the size of π . Hence, as a corollary we get:

Corollary 1. *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Int}_p}$, there exists a polynomial time algorithm in the size of π that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\mathcal{ND}_{\text{Int}_p}$.*

We point out that our technique does not require any manipulation on the proofs. We only use the fact that the proofs of the Natural Deduction calculus preserve evaluation of formulas (Lemma 1). This is not a peculiar feature of Natural Deduction calculi, but it also holds for other deductive systems for Int_p such as the Sequent Calculus of [3]. Thus the results of our paper can be restated also for different calculi. We also notice that the result of [3] is based on an implicit extraction calculus using the extraction rules cut and weakening. In this sense our result is an improvement of the one of [3], since SLD_p provides a better search strategy.

Finally, we point out that also the calculus $\text{ID}(\text{SLD}_p, [\pi])$ has the disjunction property, that is, for every $\vdash A \vee B$ provable in $\text{ID}(\text{SLD}_p, [\pi])$, either $\vdash A$ or $\vdash B$ is provable in $\text{ID}(\text{SLD}_p, [\pi])$ (see, e.g., [7]), and the proof can be found in polynomial time in the size of π .

3.1 Propositional Harrop Formulas

It is well-known that the disjunction property does not hold in general under assumptions. On the other hand it holds for sequents of the form $\Gamma \vdash A \vee B$ where Γ is a set of Harrop formulas. We recall that a propositional *Harrop formula* is either an atomic or a negated formula, or a formula of the kind $H \wedge K$, $A \rightarrow H$ where H and K are Harrop formulas and A is any formula. In [3] the authors show that if $\Gamma \vdash A \vee B$, with Γ a set of Harrop formulas, is provable in a sequent calculus for Int_p , then it can be decided in polynomial time which between the sequents $\Gamma \vdash A$ and $\Gamma \vdash B$ is intuitionistically valid.

For technical reasons, instead of considering Natural Deduction proofs with Harrop formulas as open assumptions we introduce the Harrop formulas as axioms of the calculus. To this aim, given a recursive set \mathbf{H} of Harrop formulas, we denote with $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$ the propositional Natural Deduction calculus obtained by adding the axiom-rule

$$\frac{}{\vdash H} \text{H} \in \mathbf{H}$$

to $\mathcal{ND}_{\text{Int}_p}$. Hence, if a sequent $\vdash A$ is provable in $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$, then the formula A is intuitionistically provable from the formulas in \mathbf{H} .

To treat the case of Harrop formulas in our setting, we extend the extraction calculus of the previous section with the rules RE^\wedge (*Restricted And Elimination*) and RMP (*Restricted Modus Ponens*):

$$\frac{\vdash H_1 \wedge H_2}{\vdash H_i} \text{RE}^\wedge \quad \text{with } i \in \{1, 2\} \qquad \frac{\vdash A \quad \vdash A \rightarrow K}{\vdash K} \text{RMP}$$

where H_1, H_2 and K are Harrop formulas. It is immediate to check that $\text{RE}\wedge$ and RMP are e-rules for $\mathcal{ND}_{\text{Int}_p}$. Given a recursive set of proofs Π of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$, we denote with $\text{ID}_{\text{HR}_p}([\Pi])$ the extraction calculus $\text{ID}(\{\text{SLD}_p, \text{RE}\wedge, \text{RMP}\}, [\Pi])$.

Lemma 2. *Let \mathbf{H} be a recursive set of propositional Harrop formulas and let Π be a recursive set of proofs of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$. For every Harrop formula H , if $\vdash H$ is provable in $\text{ID}_{\text{HR}_p}([\Pi])$, then $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright H$.*

Proof. The proof is by induction on the structure of H . If H is atomic or negated the assertion immediately follows. If $H \equiv H_1 \wedge H_2$ the assertion follows by the closure of $\text{ID}_{\text{HR}_p}([\Pi])$ w.r.t. the e-rule $\text{RE}\wedge$ and by the induction hypothesis. Let $H \equiv A \rightarrow K$ and suppose that $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright A$. By applying the e-rule RMP , it follows that $\vdash K$ is provable in $\text{ID}_{\text{HR}_p}([\Pi])$. Since K is a Harrop formula, by the induction hypothesis we get $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright K$. \square

Lemma 3. *Let \mathbf{H} be a recursive set of Harrop formulas and let Π be any recursive set of proofs of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$. For every proof $\pi : \Gamma \vdash A$ belonging to $[\Pi]$, if $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright \Gamma$ then $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright A$.*

Proof. The proof is similar to the one given for Lemma 1. We only have to consider the case in which π consists of an axiom-rule. In this case $\Gamma = \emptyset$ and A is a Harrop formula; since $\vdash A \in \text{Seq}([\Pi])$, it is provable in $\text{ID}_{\text{HR}_p}([\Pi])$ and, by Lemma 2, $\text{ID}_{\text{HR}_p}([\Pi]) \triangleright A$. \square

Proceeding as in the proof of Theorem 1, we get:

Theorem 3. *Let \mathbf{H} be a recursive set of Harrop formulas. If $\pi : \vdash A \vee B$ is a proof of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$, then either $\vdash A$ or $\vdash B$ is provable in $\text{ID}_{\text{HR}_p}([\pi])$.*

To study the complexity of the disjunction property we need to extend the map E_Σ of the previous section to consider the new e-rules. In this case, given a finite set of sequents Σ , $E_\Sigma : 2^{\text{SEQ}_p} \rightarrow 2^{\text{SEQ}_p}$ is defined as follows:

$$\begin{aligned} E_\Sigma(\Delta) = & \{ \vdash A \mid B_1, \dots, B_n \vdash A \in \Sigma \text{ and } \{ \vdash B_1, \dots, \vdash B_n \} \subseteq \Delta \} \\ & \cup \{ \vdash H_1 \mid H_1 \wedge H_2 \text{ is a Harrop formula and } \vdash H_1 \wedge H_2 \in \Delta \} \\ & \cup \{ \vdash H_2 \mid H_1 \wedge H_2 \text{ is a Harrop formula and } \vdash H_1 \wedge H_2 \in \Delta \} \\ & \cup \{ \vdash H \mid A \rightarrow H \text{ is a Harrop formula and } \{ \vdash A \rightarrow H, \vdash A \} \subseteq \Delta \} \end{aligned}$$

Also in this case E_Σ is a monotone and continuous operator, hence it has the least fixpoint $E_\Sigma^\infty = \bigcup_{k \in \omega} E_\Sigma^k$. It is immediate to check that $\vdash A \in E_\Sigma^\infty$, where $\Sigma = \text{Seq}([\Pi])$, iff $\vdash A$ is provable in $\text{ID}_{\text{HR}_p}([\Pi])$.

Theorem 4. *Let \mathbf{H} be a recursive set of Harrop formulas. Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$, there exists a polynomial time algorithm that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\text{ID}_{\text{HR}_p}([\pi])$.*

Proof. The proof proceeds along the lines of the proof of Theorem 2. The only difference concerns the number of iterations to get the fixpoint. As a matter of fact, it may happen that applying the operator E_Σ (where $\Sigma = \text{Seq}([\pi])$) to a set E_Σ^i no sequent of the form $\Gamma \vdash A$ of Σ is used, but only the e-rules $\text{RE}\wedge$ and RMP are applied. On the other hand, the e-rules $\text{RE}\wedge$ and RMP give rise to formulas of lower complexity than the ones in hand; thus, there is a polynomial bound (which depends on the size of π) on the number of successive applications of such rules. Hence, we need a polynomial number of iterations to get the fixpoint and each iteration requires polynomial time; this proves the assertion. \square

Since the e-rules SLD_p , $\text{RE}\wedge$ and RMP are polynomially simulable in $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$, every proof of $\text{ID}_{\text{HR}_p}([\pi])$ can be translated into a proof of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$ in polynomial time in the size of π . Hence, as a corollary we get:

Corollary 2. *Let \mathbf{H} be a recursive set of Harrop formulas and let $\pi : \vdash A \vee B$ be a proof of $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$. There exists a polynomial time algorithm that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\mathcal{ND}_{\text{Int}_p}(\mathbf{H})$.*

4 Predicate Intuitionistic Logic

Hereafter we treat first-order languages. In this case we need to consider, besides the disjunction property also the explicit definability property (ED). Some results on the complexity of these properties for Intuitionistic Logic are already given in [2], where it is shown that for first-order languages without function symbols (ED) can be decided in exponential time in the size of a proof of $\exists x A(x)$. A superexponential time algorithm is provided for the full language and it is also proved that, in this case, there exists a superexponential lower-bound.

In this section we apply our technique to Intuitionistic Logic, while in the next section we apply it to some superintuitionistic systems. Here we provide an exponential algorithm for the case of first-order languages without function symbols. A superexponential time algorithm for the full language can be obtained following the lines of [2].

In the case of first-order logic, the main extraction rule is

$$\frac{\vdash \theta A_1 \quad \dots \quad \vdash \theta A_n \quad \overline{A_1, \dots, A_n \vdash B}}{\vdash \theta B} \text{SLD}$$

where A_1, \dots, A_n, B are arbitrary formulas and θ is any substitution. Given a set of proofs Π of a calculus \mathbf{C} we denote with \mathcal{L}_Π the restriction of the language \mathcal{L} to the formulas built over the individual variables, the constant symbols and the predicate symbols occurring in $[\Pi]$; if $[\Pi]$ does not contain any individual variable, we add the variable x to \mathcal{L}_Π .

Definition 3 (First-Order Evaluation). *Let Π be a set of proofs of a calculus \mathbf{C} over a language \mathcal{L} and let A be a formula of \mathcal{L} . A is evaluated in Π (in symbols $\Pi \triangleright A$) iff the following conditions hold:*

- (i). *There exists a proof $\pi : \vdash A \in \Pi$.*
- (ii). *One of the following inductive conditions holds:*
 - a) *A is an atomic or a negated formula;*
 - b) *$A \equiv B \wedge C$ and $\Pi \triangleright B$ and $\Pi \triangleright C$;*
 - c) *$A \equiv B \vee C$ and either $\Pi \triangleright B$ or $\Pi \triangleright C$;*
 - d) *$A \equiv B \rightarrow C$ and if $\Pi \triangleright B$, then $\Pi \triangleright C$;*
 - e) *$A \equiv \forall x B(x)$ and $\Pi \triangleright B(t)$ for every term t of \mathcal{L}_Π ;*
 - f) *$A \equiv \exists x B(x)$ and $\Pi \triangleright B(t)$ for some term t of \mathcal{L}_Π .*

A set Γ of formulas is evaluated in Π (and we write $\Pi \triangleright \Gamma$) if $\Pi \triangleright A$ holds for every $A \in \Gamma$. Given a set Π of proofs of a calculus \mathbf{C} , the *closure under substitution* of $[\Pi]$ is the set containing the proof $\theta\pi$ for every substitution θ and every $\pi \in [\Pi]$ (we recall that $\theta\pi$ is the proof obtained by substituting every free variable x occurring in π with $\theta(x)$). The following fact can be proved:

Lemma 4. *Let Π be a recursive set of proofs of $\mathcal{ND}_{\text{Int}}$; for every $\pi : \Gamma \vdash A$ over \mathcal{L}_{Π} belonging to the closure under substitution of $[\Pi]$, if $\text{ID}(\text{SLD}, [\Pi]) \triangleright \Gamma$ then $\text{ID}(\text{SLD}, [\Pi]) \triangleright A$.*

Proof. Since $\Gamma = \{B_1, \dots, B_n\}$ is evaluated in $\text{ID}(\text{SLD}, [\Pi])$, $\text{ID}(\text{SLD}, [\Pi])$ contains the proofs $\tau_1 : \vdash B_1, \dots, \tau_n : \vdash B_n$; moreover, there exist $\pi' : \Gamma' \vdash A' \in [\Pi]$ and a substitution θ over \mathcal{L}_{Π} such that $\pi : \Gamma \vdash A$ coincides with $\theta\pi' : \theta\Gamma' \vdash \theta A'$. This means that the proof

$$\frac{\tau_1 : \vdash \theta B'_1, \dots, \tau_n : \vdash \theta B'_n \quad \overline{B'_1, \dots, B'_n \vdash A'}}{\vdash \theta A'} \text{SLD}$$

belongs to $\text{ID}(\text{SLD}, [\Pi])$ and this proves Point (i) of Definition 3. Point (ii) is proved as in Lemma 1; we only have to discuss the predicate rules. Here we give a representative case.

Exists Elimination.

$$\pi : \Gamma \vdash A \equiv \frac{\pi_0 : \Gamma_0 \vdash \exists x B(x) \quad \pi_1 : \Gamma_1, B(y) \vdash C}{\Gamma_0, \Gamma_1 \vdash C} \text{E}\exists$$

By induction hypothesis $\text{ID}(\text{SLD}, [\Pi]) \triangleright \exists x B(x)$, therefore there exists a term t of \mathcal{L}_{Π} such that $\text{ID}(\text{SLD}, [\Pi]) \triangleright B(t)$. Let $\pi'_1 : \Gamma, B(t) \vdash C$ be the proof of \mathcal{L}_{Π} obtained by substituting y with t in π_1 (note that, being y the proper parameter of the $\text{E}\exists$ -rule, the substitution does not act on Γ_1 and on C). Since π'_1 belongs to the closure under substitution of $[\Pi]$ and $\text{ID}(\text{SLD}, [\Pi]) \triangleright \Gamma_1$, by induction hypothesis on π'_1 we can conclude $\text{ID}(\text{SLD}, [\Pi]) \triangleright C$. \square

Proceeding as in Theorem 1 one can prove:

Theorem 5. (i). *If $\pi : \vdash A \vee B$ belongs to $\mathcal{ND}_{\text{Int}}$, then either $\vdash A$ or $\vdash B$ is provable in $\text{ID}(\text{SLD}, [\Pi])$;*
(ii). *If $\pi : \vdash \exists x A(x)$ belongs to $\mathcal{ND}_{\text{Int}}$, then there exists a term t of $\mathcal{L}_{\{\pi\}}$ such that $\vdash A(t)$ is provable in $\text{ID}(\text{SLD}, [\Pi])$.*

Given a proof π of $\mathcal{ND}_{\text{Int}}$, let

$$\text{Seq}^*([\pi]) = \{\theta\Gamma \vdash \theta A \mid \Gamma \vdash A \in \text{Seq}([\pi]) \text{ and } \theta \text{ is a substitution over } \mathcal{L}_{\{\pi\}}\}$$

Clearly, the cardinality of $\text{Seq}^*([\pi])$ is exponential in the size of π . Let E_{Σ}^{∞} be defined as in Section 3; it is easy to check that, for all θ over $\mathcal{L}_{\{\pi\}}$, $\vdash \theta A \in E_{\Sigma}^{\infty}$ iff $\vdash \theta A$ is provable in $\text{ID}(\text{SLD}, [\pi])$, where $\Sigma = \text{Seq}^*([\pi])$. Applying the same reasoning of Theorem 2, one can check that the number of iterations required to build the fixpoint E_{Σ}^{∞} is linear in the cardinality of $\text{Seq}^*([\pi])$ and that any iteration can be accomplished in polynomial time in $|\text{Seq}^*([\pi])|$. Therefore:

Theorem 6. (i). *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Int}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\text{ID}(\text{SLD}, [\pi])$.*

- (ii). Given a proof $\pi : \vdash \exists xA(x)$ in $\mathcal{ND}_{\text{Int}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A(t)$ in $\text{ID}(\text{SLD}, [\pi])$.

We remark that also in this case every proof of $\text{ID}(\text{SLD}, [\pi])$ can be translated into a proof of $\mathcal{ND}_{\text{Int}}$ in polynomial time in the size of π . The same remark also holds for the analogous results given in the next sections.

4.1 Harrop Formulas

In the first-order setting a *Harrop formula* is either an atomic or a negated formula, or a formula of the kind $H \wedge K$, $A \rightarrow H$, $\forall xH$, where H and K are Harrop formulas and A is any formula. To treat the case of Harrop formulas at the predicate level, we need to consider besides the e-rule SLD and the e-rules $\text{RE}\wedge$ and RMP of Section 3.1, the e-rule $\text{RE}\forall$ (*Restricted For-All Elimination*)

$$\frac{\vdash \forall xH(x)}{\vdash H(t)} \text{RE}\forall$$

where $\forall xH(x)$ is a Harrop formula and t is any term of \mathcal{L} . Given a finite set of sequents Σ over \mathcal{L} , let \mathcal{L}_Σ be the language containing only the constant symbols, the individual variables and the predicate symbols occurring in Σ . The function $E_\Sigma : 2^{\text{SEQ}} \rightarrow 2^{\text{SEQ}}$ is defined as follows:

$$\begin{aligned} E_\Sigma(\Delta) = & \{ \vdash A \mid B_1, \dots, B_n \vdash A \in \Sigma \text{ and } \{ \vdash B_1, \dots, \vdash B_n \} \subseteq \Delta \} \\ & \cup \{ \vdash H_1 \mid H_1 \wedge H_2 \text{ is a Harrop formula and } \vdash H_1 \wedge H_2 \in \Delta \} \\ & \cup \{ \vdash H_2 \mid H_1 \wedge H_2 \text{ is a Harrop formula and } \vdash H_1 \wedge H_2 \in \Delta \} \\ & \cup \{ \vdash H \mid A \rightarrow H \text{ is a Harrop formula and } \{ \vdash A \rightarrow H, \vdash A \} \subseteq \Delta \} \\ & \cup \{ \vdash H(t) \mid \forall xH(x) \text{ is a Harrop formula, } t \text{ is any term of } \mathcal{L}_\Sigma \text{ and} \\ & \quad \vdash \forall xH(x) \in \Delta \} \end{aligned}$$

Given a recursive set of Harrop formulas \mathbf{H} and a recursive set of proofs Π of $\mathcal{ND}_{\text{Int}}(\mathbf{H})$, we denote with $\text{ID}_{\mathbf{HR}}([\Pi])$ the extraction calculus

$$\text{ID}(\{\text{SLD}, \text{RE}\wedge, \text{RMP}, \text{RE}\forall\}, [\Pi])$$

Lemma 2 can be easily extended as follows:

Lemma 5. *Let \mathbf{H} be a recursive set of Harrop formulas and let Π be a recursive set of proofs of $\mathcal{ND}_{\text{Int}}(\mathbf{H})$. For every Harrop formula H , if $\vdash H$ is provable in $\text{ID}_{\mathbf{HR}}([\Pi])$, then $\text{ID}_{\mathbf{HR}}([\Pi]) \triangleright H$.*

Following the lines of Section 3.1 and of Theorem 6 one can prove:

Theorem 7. *Let \mathbf{H} be a recursive set of Harrop formulas.*

- (i). *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Int}}(\mathbf{H})$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\text{ID}_{\mathbf{HR}}([\pi])$.*
- (ii). *Given a proof $\pi : \vdash \exists xA(x)$ in $\mathcal{ND}_{\text{Int}}(\mathbf{H})$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A(t)$ in $\text{ID}_{\mathbf{HR}}([\pi])$.*

5 Intermediate Logics

An *intermediate logic* is any set of formulas \mathbf{L} such that $\mathbf{Int} \subseteq \mathbf{L} \subseteq \mathbf{Cl}$ (where \mathbf{Cl} denotes the set of classically valid formulas) and \mathbf{L} is closed under modus ponens, generalization and predicate substitution (see, e.g., [14] for a detailed definition).

5.1 Kuroda Logic

In this section we treat the case of Kuroda Logic, the Intermediate Logic obtained by adding to Intuitionistic Logic the axiom-schema

$$(\text{Kur}) \equiv \forall x \neg \neg A(x) \rightarrow \neg \neg \forall x A(x)$$

This principle has been deeply investigated in the literature on constructive systems, see e.g. [9,16]. Moreover, this principle has been considered in the context of Abstract Data Types specification based on *isoinitial (classical) semantics* for the role it plays with respect to Classical Logic (see [12,13]). Indeed, a theory \mathbf{T} is classically consistent iff it is consistent with respect to any intermediate predicate logic \mathbf{L} including Kuroda Principle [9].

A Natural Deduction calculus $\mathcal{ND}_{\mathbf{Kur}}$ for Kuroda Logic can be obtained by adding the rule

$$\frac{\Gamma \vdash \forall x \neg \neg A(x)}{\Gamma \vdash \neg \neg \forall x A(x)} \text{Kur}$$

to $\mathcal{ND}_{\mathbf{Int}}$. To treat Kuroda Logic we do not need to add new e-rules to the extraction calculus. As a matter of fact Lemma 4 holds for any recursive set of proofs Π of $\mathcal{ND}_{\mathbf{Kur}}$, since the rule Kur introduces a sequent whose consequence is a negated formula. Hence, proceeding as in the previous section we get:

- Theorem 8.** (i). *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\mathbf{Kur}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\mathbf{ID}(\text{SLD}, [\pi])$.*
- (ii). *Given a proof $\pi : \vdash \exists x A(x)$ in $\mathcal{ND}_{\mathbf{Kur}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A(t)$ in $\mathbf{ID}(\text{SLD}, [\pi])$.*

5.2 Grzegorzcyk Logic

Grzegorzcyk Logic is the Intermediate Logic obtained by adding to Intuitionistic Logic the axiom-schema

$$(\text{Grz}) \equiv \forall x (A(x) \vee B) \rightarrow \forall x A(x) \vee B \quad \text{where } x \text{ is not free in } B$$

This logic is characterized by the class of Kripke models with constant domains (see [8,9,10,15]). A Natural Deduction calculus $\mathcal{ND}_{\mathbf{Grz}}$ for this logic is obtained by adding to $\mathcal{ND}_{\mathbf{Int}}$ the rule

$$\frac{\Gamma \vdash \forall x(A(x) \vee B)}{\Gamma \vdash \forall x A(x) \vee B} \text{Grz} \quad \text{where } x \text{ is not free in } B$$

To treat this logic we need the following e-rule (a restricted version of \forall introduction)

$$\frac{\vdash A(x) \quad \vdash \forall x A(x) \vee B}{\vdash \forall x A(x)} \text{RGrz}$$

Given a recursive set of proofs Π of $\mathcal{ND}_{\text{Grz}}$, $\text{ID}_{\text{Grz}}([\Pi])$ denotes the extraction calculus $\text{ID}(\{\text{SLD}, \text{RGrz}\}, [\Pi])$.

Lemma 6. *Let Π be any recursive set of proofs of $\mathcal{ND}_{\text{Grz}}$. For every proof $\pi : \Gamma \vdash A$ over \mathcal{L}_{Π} belonging to the closure under substitution of $[\Pi]$, if $\text{ID}_{\text{Grz}}([\Pi]) \triangleright \Gamma$ then $\text{ID}_{\text{Grz}}([\Pi]) \triangleright A$.*

Proof. Since $\text{ID}_{\text{Grz}}([\Pi])$ contains the SLD rule, it is immediate to check that $\vdash A$ is provable in $\text{ID}_{\text{Grz}}([\Pi])$. The proof of Point (ii) of Definition 3 is, as usual, by induction on $\text{depth}(\pi)$, the only difference concerns the treatment of the rule Grz.

Rule Grz.

$$\pi : \Gamma \vdash A \equiv \frac{\pi_1 : \Gamma \vdash \forall x(B(x) \vee C)}{\Gamma \vdash \forall x B(x) \vee C} \text{Grz}$$

If $\text{ID}_{\text{Grz}}([\Pi]) \triangleright C$ the assertion immediately holds. Let us assume that C is not evaluated in $\text{ID}_{\text{Grz}}([\Pi])$. If $\text{ID}_{\text{Grz}}([\Pi]) \triangleright \Gamma$, by induction hypothesis applied to the proof π_1 , we get that $\text{ID}_{\text{Grz}}([\Pi]) \triangleright \forall x(B(x) \vee C)$; this means that there exists a proof $\tau : \vdash \forall x(B(x) \vee C)$ in $\text{ID}_{\text{Grz}}([\Pi])$ and, for every term t of the language \mathcal{L}_{Π} , $\text{ID}_{\text{Grz}}([\Pi]) \triangleright B(t/x)$. It only remains to show that $\text{ID}_{\text{Grz}}([\Pi])$ contains a proof of $\vdash \forall x B(x)$. Let x be an individual variable of \mathcal{L}_{Π} (such a variable always exists by definition of \mathcal{L}_{Π}); then $\text{ID}_{\text{Grz}}([\Pi]) \triangleright B(x)$, hence there exists a proof $\tau' : \vdash B(x)$ in $\text{ID}_{\text{Grz}}([\Pi])$. Applying the e-rule RGrz to τ and τ' , we get the required proof. \square

Given a finite set of sequents Σ , we extend the function $E_{\Sigma} : 2^{\text{SEQ}} \rightarrow 2^{\text{SEQ}}$ as follows:

$$E_{\Sigma}(\Delta) = \{ \vdash A \mid B_1, \dots, B_n \vdash A \in \Sigma \text{ and } \{ \vdash B_1, \dots, \vdash B_n \} \subseteq \Delta \} \\ \cup \{ \vdash \forall x A(x) \mid \{ \vdash A(x), \vdash \forall x(A(x) \vee B) \} \subseteq \Delta \}$$

Also in this case $E_{\Sigma}^{\infty} = \bigcup_{k \in \omega} E_{\Sigma}^k$ and, for every θ over $\mathcal{L}_{\{\pi\}}$, $\vdash \theta A \in E_{\Sigma}^{\infty}$ iff $\vdash \theta A$ is provable in $\text{ID}_{\text{Grz}}([\pi])$, where $\Sigma = \text{Seq}^*([\pi])$. Now, applying the same reasoning of Theorem 2, one can check that the number of iterations required to build the fixpoint E_{Σ}^{∞} is linear in the cardinality of $\text{Seq}^*([\pi])$ and that any iteration can be accomplished in polynomial time in $|\text{Seq}^*([\pi])|$. Therefore:

Theorem 9. (i). *Given a proof $\pi : \vdash A \vee B$ in $\mathcal{ND}_{\text{Grz}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A$ or a proof of $\vdash B$ in the calculus $\text{ID}_{\text{Grz}}([\pi])$.*

(ii). *Given a proof $\pi : \vdash \exists x A(x)$ in $\mathcal{ND}_{\text{Grz}}$, there exists an exponential time algorithm in the size of π that constructs a proof of $\vdash A(t)$ in $\text{ID}_{\text{Grz}}([\pi])$.*

5.3 Harrop Formulas

To conclude this section we notice that Theorem 7 can be also extended to proofs of the calculi $\mathcal{ND}_{\mathbf{Kur}}(\mathbf{H})$ for Kuroda Logic and $\mathcal{ND}_{\mathbf{Grz}}(\mathbf{H})$ for Grzegorczyk Logic using Harrop formulas as axioms. In the former case one has to use the extraction calculus $\mathbf{ID}_{\mathbf{HR}}([\pi])$ defined in Section 4.1, in the latter one has to extend such a calculus with the e-rule \mathbf{rGrz} . For the sake of completeness, we also remark that the same result can be formulated also for proofs of a natural deduction calculus containing both the rules \mathbf{Kur} and \mathbf{Grz} .

6 Conclusions and Future Work

In this paper we have shown that the problem to decide (DP) and (ED) in Intuitionistic Logic, Kuroda Logic and Grzegorczyk Logic, with and without Harrop formulas as axioms, has the same complexity. However, there are some propositional intermediate logics that can be treated in our framework, for which we are not able to give polynomial time algorithms to decide (DP), while we can exhibit exponential time algorithms. Among these we mention the well-known *Kreisel-Putnam Logic* [11], obtained by adding to \mathbf{Int}_p the axiom-schema

$$(\neg A \rightarrow B \vee C) \rightarrow (\neg A \rightarrow B) \vee (\neg A \rightarrow C)$$

and *Scott Logic* [11], obtained by adding to \mathbf{Int}_p the axiom-schema

$$((\neg\neg A \rightarrow A) \rightarrow \neg A \vee \neg\neg A) \rightarrow \neg A \vee \neg\neg A$$

We remark that also for these logics (DP) can be decided using purely intuitionistic extraction rules (a proof can be found in [4,7]), but the resulting algorithms are exponential in the size of the proof. We consider an interesting question to further investigate the complexity of (DP) for these logics.

References

1. A. Avellone, M. Ferrari, and C. Fiorentini. A formal framework for synthesis and verification of logic programs. In K.-K. Lau, editor, *Logic Based Program Synthesis and Transformation, 10th International Workshop, LOPSTR 2000, Selected Papers*, volume 2042 of *Lecture Notes in Computer Science*, pages 1–17. Springer-Verlag, 2001.
2. S. Buss and G. Mints. The complexity of the disjunction and existential properties in intuitionistic logic. *Annals of Pure and Applied Logic*, 99(3):93–104, 1999.
3. S. Buss and P. Pudlák. On the computational content of intuitionistic propositional proofs. *Annals of Pure and Applied Logic*, 109(1-2):49–64, 2001.
4. M. Ferrari. *Strongly Constructive Formal Systems*. PhD thesis, Dipartimento di Scienze dell’Informazione, Università degli Studi di Milano, Italy, 1997. Available at <http://homes.dsi.unimi.it/~ferram>.
5. M. Ferrari and C. Fiorentini. A proof-theoretical analysis of semiconstructive intermediate theories. *Studia Logica*, to appear.

6. M. Ferrari, C. Fiorentini, and P. Miglioli. Goal oriented information extraction in uniformly constructive calculi. In *Argentinian Workshop on Theoretical Computer Science (WAIT'99)*, pages 51–63. Sociedad Argentina de Informática e Investigación Operativa, 1999.
7. M. Ferrari, P. Miglioli, and M. Ornaghi. On uniformly constructive and semiconstructive formal systems. *Logic Journal of the IGPL*, to appear.
8. C. Fiorentini and P. Miglioli. A cut-free sequent calculus for the logic of constant domains with a limited amount of duplications. *Logic Journal of the IGPL*, 7(6):733–753, 1999.
9. D.M. Gabbay. *Semantical Investigations in Heyting's Intuitionistic Logic*. Reidel, Dordrecht, 1981.
10. S. Görnemann. A logic stronger than intuitionism. *Journal of Symbolic Logic*, 36:249–261, 1971.
11. G. Kreisel and H. Putnam. Eine Unableitbarkeitsbeweismethode für den intuitionistischen Aussagenkalkül. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 3:74–78, 1957.
12. P. Miglioli, U. Moscato, and M. Ornaghi. Constructive theories with abstract data types for program synthesis. In D.G. Skordev, editor, *Mathematical Logic and its Applications*, pages 293–302. Plenum Press, New York, 1987.
13. P. Miglioli, U. Moscato, and M. Ornaghi. Abstract parametric classes and abstract data types defined by classical and constructive logical methods. *The Journal of Symbolic Computation*, 18(1):41–81, 1994.
14. H. Ono. Some results on the intermediate logics. *Publications of the Research Institute for Mathematical Sciences, Kyoto University*, 8:117–130, 1972.
15. C.A. Smorynski. Applications of Kripke semantics. In A.S. Troelstra, editor, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*, pages 324–391. Springer-Verlag, 1973.
16. A.S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1973.
17. A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.